Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

SEN

Software Engineering

Software ENgineering

Reducing dynamic epistemic logic to PDL by program transformation

D.J.N. van Eijck

REPORT SEN-E0423 DECEMBER 2004

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-369X

Reducing dynamic epistemic logic to PDL by program transformation

ABSTRACT
We present a direct reduction of dynamic epistemic logic in the spirit of Baltag, Moss and Solecki to propositional dynamic logic (PDL) by program transformation. The program transformation approach associates with every update action a transformation on PDL programs. These transformations are then employed in reduction axioms for the update actions. It follows that the logic of public announcement, the logic of group announcements, the logic of secret message passing, and so on, can all be viewed as subsystems of PDL. Moreover, the program transformation approach can be used to generate the appropriate reduction axioms for these logics. Our direct reduction of dynamic epistemic logic to PDL was inspired by the reduction of dynamic epistemic logic to automata PDL of Van Benthem and Kooi Our approach shows how the detour through automata can be avoided.

1998 ACM Computing Classification System: E 4, F 4.1, H 1.1
Keywords and Phrases: Dynamic epistemic logic, logic of communication, propositional dynamic logic, program transformation.

Reducing Dynamic Epistemic Logic to PDL by Program Transformation

Jan van Eijck
CWI and ILLC, Amsterdam, Uil-OTS, Utrecht

December 7, 2004

Abstract

We present a direct reduction of dynamic epistemic logic in the spirit of [4] to propositional dynamic logic (PDL) [17, 18] by program transformation. The program transformation approach associates with every update action a transformation on PDL programs. These transformations are then employed in reduction axioms for the update actions. It follows that the logic of public announcement, the logic of group announcements, the logic of secret message passing, and so on, can all be viewed as subsystems of PDL. Moreover, the program transformation approach can be used to generate the appropriate reduction axioms for these logics. Our direct reduction of dynamic epistemic logic to PDL was inspired by the reduction of dynamic epistemic logic to automata PDL of [13]. Our approach shows how the detour through automata can be avoided.

1 Introduction

Dynamic epistemic logic [1, 2, 3, 4] analyses the changes in epistemic information among sets of agents that result from various communicative actions, such as public announcements, group messages and individual messages. The logics studied in [4] add information update operations to epistemic description languages with a common knowledge operator, in such a way that the addition increases expressive power. In [13] it is demonstrated how update axioms can be made susceptible to reduction axioms, by the simple means of switching to more expressive epistemic description languages. More in particular, it is shown in [13] how generic updates with epistemic actions can be axiomatized in automata PDL [12, Chapter 10.3].

Knowledge of the reduction of dynamic epistemic logic to automata PDL from [13], combined with the fact that automata PDL and standard PDL obviously have the same expressive power (obviously, for every finite automaton generates a regular language, and PDL is the logic of regular programs), kindles a natural desire for a direct reduction of DEL to PDL, a desire we will satisfy below. We will show that the detour through automata is unnecessary, by demonstrating how the effects of generic updating can be captured by PDL transformations of PDL programs. In terms of these transformations a straightforward axiomatisation of dynamic update logic in PDL is achieved.

2 PDL and Updates

Let p range over a set of basic propositions P and let a range over a set of agents Ag. Then the language of PDL over P, Ag is given by:

$$\begin{array}{rcl} \varphi & ::= & \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi \\ \pi & ::= & a \mid ?\varphi \mid \pi_1;\pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^* \end{array}$$

Employ the usual abbreviations: $\bot$ is shorthand for $\neg\top$, $\varphi_1 \vee \varphi_2$ is shorthand for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ is shorthand for $\neg(\varphi_1 \wedge \varphi_2)$, $\varphi_1 \leftrightarrow \varphi_2$ is shorthand for $(\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$, and $\langle\pi\rangle\varphi$ is shorthand for $\neg[\pi]\neg\varphi$. Also, if $B \subseteq Ag$ and B is finite, use B as shorthand for $b_1 \cup b_2 \cup \cdots$. Under this convention, the general knowledge operator $E_B\varphi$ takes the shape $[B]\varphi$, while the common knowledge operator $C_B\varphi$ appears as $[B^*]\varphi$, i.e., $[B]\varphi$ expresses that it is general knowledge among agents B that φ, and $[B^*]\varphi$ expresses that it is common knowledge among agents B that φ. In the special case where $B = \emptyset$, B turns out equivalent to $?\bot$, the program that always fails.

The semantics of PDL over P, Ag is given relative to labelled transition systems $\mathbf{M} = (W, V, R)$, where W is a set of worlds (or states), $V : W \rightarrow \mathcal{P}(P)$ is a valuation function, and $R = \{\stackrel{a}{\rightarrow} \subseteq W \times W \mid a \in Ag\}$ is a set of labelled transitions, i.e., binary relations on W, one for each label a. In what follows, we will take the labeled transitions for a to represent the epistemic alternatives of an agent a.

The formulae of PDL are interpreted as subsets of $W_{\mathbf{M}}$ (the state set of $\mathbf{M}$), the actions of PDL as binary relations on $W_{\mathbf{M}}$, as follows:

$$\begin{array}{rcl} [\![\top]\!]^{\mathbf{M}} & = & W_{\mathbf{M}} \\ [\![p]\!]^{\mathbf{M}} & = & \{w \in W_{\mathbf{M}} \mid p \in V_{\mathbf{M}}(w)\} \\ [\![\neg\varphi]\!]^{\mathbf{M}} & = & W_{\mathbf{M}} - [\![\varphi]\!]^{\mathbf{M}} \\ [\![\varphi_1 \wedge \varphi_2]\!]^{\mathbf{M}} & = & [\![\varphi_1]\!]^{\mathbf{M}} \cap [\![\varphi_2]\!]^{\mathbf{M}} \\ [\![[\pi]\varphi]\!]^{\mathbf{M}} & = & \{w \in W_{\mathbf{M}} \mid \forall v(\text{ if } (w,v) \in [\![\pi]\!]^{\mathbf{M}} \text{ then } v \in [\![\varphi]\!]^{\mathbf{M}})\} \\ \\ [\![a]\!]^{\mathbf{M}} & = & \stackrel{a}{\rightarrow}_{\mathbf{M}} \\ [\![?\varphi]\!]^{\mathbf{M}} & = & \{(w,w) \in W_{\mathbf{M}} \times W_{\mathbf{M}} \mid w \in [\![\varphi]\!]^{\mathbf{M}}\} \\ [\![\pi_1;\pi_2]\!]^{\mathbf{M}} & = & [\![\pi_1]\!]^{\mathbf{M}} \circ [\![\pi_2]\!]^{\mathbf{M}} \\ [\![\pi_1 \cup \pi_2]\!]^{\mathbf{M}} & = & [\![\pi_1]\!]^{\mathbf{M}} \cup [\![\pi_2]\!]^{\mathbf{M}} \\ [\![\pi^*]\!]^{\mathbf{M}} & = & ([\![\pi]\!]^{\mathbf{M}})^* \end{array}$$

If $w \in W_{\mathbf{M}}$ then we use $\mathbf{M} \models_w \varphi$ for $w \in [\![\varphi]\!]^{\mathbf{M}}$.

[4] proposes to model epistemic actions as epistemic models, with valuations replaced by preconditions. See also: [5, 6, 7, 8, 9, 11, 14, 19].

Action models for a given language $\mathcal{L}$ Let a set of agents Ag and an epistemic language $\mathcal{L}$ be given. An action model for $\mathcal{L}$ is a triple $A = ([s_0, \ldots, s_{n-1}], \text{pre}, T)$ where $[s_0, \ldots, s_{n-1}]$ is

a finite list of action states, pre : $\{s_0, \ldots, s_{n-1}\} \to \mathcal{L}$ assigns a precondition to each action state, and $T : Ag \to \mathcal{P}(\{s_0, \ldots, s_{n-1}\}^2)$ assigns an accessibility relation $\stackrel{a}{\to}$ to each agent $a \in Ag$.

A pair $\mathbf{A} = (A, s)$ with $s \in \{s_0, \ldots, s_{n-1}\}$ is a pointed action model, where s is the action that actually takes place.

The list ordering of the action states in an action model will play an important role in the definition of the program transformations associated with the action models.

In the definition of action models, $\mathcal{L}$ can be any language that can be interpreted in PDL models. Actions can be executed in PDL models by means of the following product construction:

Action Update Let a PDL model $\mathbf{M} = (W, V, R)$, a world $w \in W$, and a pointed action model (A, s), with $A = ([s_0, \ldots, s_{n-1}], \mathrm{pre}, T)$, be given. Then the result of executing (A, s) in $(\mathbf{M}, w)$ is the model $(\mathbf{M} \otimes A, (w, s))$, with $\mathbf{M} \otimes A = (W', V', R')$, where

$$\begin{aligned} W' &= \{(w, s) \mid s \in \{s_0, \ldots, s_{n-1}\}, w \in [\![\mathrm{pre}(s)]\!]^{\mathbf{M}}\} \\ V'(w, s) &= V(w) \\ R'(a) &= \{((w, s), (w', s')) \mid (w, w') \in R(a), (s, s') \in T(a)\}. \end{aligned}$$

The language of $\mathrm{PDL}^{\mathrm{DEL}}$ (update PDL) is given by extending the PDL language with update constructions $[A, s]\varphi$, where (A, s) is a pointed action model. The interpretation of $[A, s]\varphi$ in $\mathbf{M}$ is given by:

$$[\![[A, s]\varphi]\!]^{\mathbf{M}} = \{w \in W_{\mathbf{M}} \mid \text{if } \mathbf{M} \models_w \mathrm{pre}(s) \text{ then } (w, s) \in [\![\varphi]\!]^{\mathbf{M} \otimes A}\}.$$

Using $\langle A, s\rangle\varphi$ as shorthand for $\neg[A, s]\neg\varphi$, we see that the interpretation for $\langle A, s\rangle\varphi$ turns out as:

$$[\![\langle A, s\rangle\varphi]\!]^{\mathbf{M}} = \{w \in W_{\mathbf{M}} \mid \mathbf{M} \models_w \mathrm{pre}(s) \text{ and } (w, s) \in [\![\varphi]\!]^{\mathbf{M} \otimes A}\}.$$

Updating with multiple pointed update actions is also possible. A multiple pointed action is a pair (A, S), with A an action model, and S a subset of the state set of A. Extend the language with updates $[A, S]\varphi$, and interpret this as follows:

$$[\![[A, S]\varphi]\!]^{\mathbf{M}} = \{w \in W_{\mathbf{M}} \mid \forall s \in S(\text{ if } \mathbf{M} \models_w \mathrm{pre}(s) \text{ then } \mathbf{M} \otimes A \models_{(w,s)} \varphi)\}.$$

In what follows we will concentrate on updates with (single) pointed action models.

3 Program Transformation

We will now show how $\mathrm{PDL}^{\mathrm{DEL}}$ formulas can be reduced to PDL formulas. For every action model A with states $s_0, \ldots, s_{n-1}$ we define a set of n^2 program transformers $T^A_{i,j}$ $(0 \leq i < n, 0 \leq$

$j < n$), as follows:

$$\begin{array}{rcl}
T^A_{ij}(a) & = & \begin{cases} ?\mathrm{pre}(s_i); a & \text{if } s_i \xrightarrow{a} s_j, \\ ?\bot & \text{otherwise} \end{cases} \\
T^A_{ij}(?\varphi) & = & \begin{cases} ?(\mathrm{pre}(s_i) \wedge [A, s_i]\varphi) & \text{if } i = j, \\ ?\bot & \text{otherwise} \end{cases} \\
T^A_{ij}(\pi_1; \pi_2) & = & \bigcup_{k=0}^{n-1} (T^A_{ik}(\pi_1); T^A_{kj}(\pi_2)) \\
T^A_{ij}(\pi_1 \cup \pi_2) & = & T^A_{ij}(\pi_1) \cup T^A_{ij}(\pi_2) \\
T^A_{ij}(\pi^*) & = & K^A_{ijn}(\pi)
\end{array}$$

where $K^A_{ijk}(\pi)$ is a (transformed) program for all the π^* paths from s_i to s_j that can be traced through A while avoiding a pass through intermediate states s_k and higher.

In particular:

- $K^A_{ij0}(\pi)$ is a program for all the π^* paths from s_i to s_j that can be traced through A without stopovers at intermediate states, i.e., if $i = j$ it either is the skip action or a direct π loop, and otherwise it is a direct π step.
- $K^A_{ijn}(\pi)$ is a program for all the π^* paths from s_i to s_j that can be traced through A, for stopovers at any s_k ($0 \leq k \leq n-1$) are allowed.

Note that it is immaterial *how many times* a stopover is made at a particular intermediate state.

$K^A_{ijk}(\pi)$ is defined by recursing on k, as follows:

$$K^A_{ij0}(\pi) = \begin{cases} ?\top \cup T^A_{ij}(\pi) & \text{if } i = j, \\ T^A_{ij}(\pi) & \text{otherwise} \end{cases}$$

$$K^A_{ij(k+1)}(\pi) = \begin{cases} (K^A_{kkk}(\pi))^* & \text{if } i = k = j, \\ (K^A_{kkk}(\pi))^*; K^A_{kjk}(\pi) & \text{if } i = k \neq j, \\ K^A_{ikk}(\pi); (K^A_{kkk}(\pi))^* & \text{if } i \neq k = j, \\ K^A_{ijk}(\pi) \cup (K^A_{ikk}(\pi); (K^A_{kkk}(\pi))^*; K^A_{kjk}(\pi)) & \text{otherwise } (i \neq k \neq j). \end{cases}$$

For some runs through example applications of these definitions, see Section 5 below.

Lemma 1 (Kleene Path) *Suppose* $(w, w') \in [\![T^A_{ij}(\pi)]\!]^{\mathbf{M}}$ *iff there is a* π *path from* (w, s_i) *to* (w', s_j) *in* $\mathbf{M} \otimes A$. *Then* $(w, w') \in [\![K^A_{ijn}(\pi)]\!]^{\mathbf{M}}$ *iff there is a* π^* *path from* (w, s_i) *to* (w', s_j) *in* $\mathbf{M} \otimes A$.

Proof. Use the definition of K^A_{ijk} to prove by induction on k that $(w, w') \in [\![K^A_{ijk}(\pi)]\!]^{\mathbf{M}}$ iff there is a π^* path from (w, s_i) to (w', s_j) in $\mathbf{M} \otimes A$ that does not pass through any pairs (v, s) with $s \in \{s_k, \ldots, s_{n-1}\}$.

Base case, $i = j$: A π^* path from (w, s_i) to (w', s_j) that does not visit any intermediate states is either the empty path or a single π step from (w, s_i) to (w', s_j). Such a path exists iff $(w, w') \in [\![?\top \cup T^A_{ij}]\!]^{\mathbf{M}}$ iff $(w, w') \in [\![K^A_{ij0}(\pi)]\!]^{\mathbf{M}}$.

Base case, $i \neq j$: A π^* path from (w, s_i) to (w', s_j) that does not visit any intermediate states is a single π step from (w, s_i) to (w', s_j). Such a path exists iff $(w, w') \in [\![T^A_{ij}]\!]^{\mathbf{M}}$ iff $(w, w') \in [\![K^A_{ij0}(\pi)]\!]^{\mathbf{M}}$.

Induction step. Assume that $(w, w') \in [\![K^A_{ijk}(\pi)]\!]^{\mathbf{M}}$ iff there is a π^* path from (w, s_i) to (w', s_j) in $\mathbf{M} \otimes A$ that does not pass through any pairs (v, s) with $s \in \{s_k, \ldots, s_{n-1}\}$.

Case $i = k = j$. A path from (w, s_i) to (w', s_j) in $\mathbf{M} \otimes A$ that does not pass through any pairs (v, s) with $s \in \{s_{k+1}, \ldots, s_{n-1}\}$ now consists of an arbitrary number of π^* paths from s_k to s_k that do not visit any intermediate states with action component s_k or higher. By the induction hypothesis, such a path exists iff $(w, w') \in [\![(K^A_{kkk}(\pi))^*]\!]^{\mathbf{M}}$ iff $(w, w') \in [\![K^A_{ij(k+1)}(\pi)]\!]^{\mathbf{M}}$.

Case $i = k \neq j$. A path from (w, s_i) to (w', s_j) in $\mathbf{M} \otimes A$ that does not pass through any pairs (v, s) with $s \in \{s_{k+1}, \ldots, s_{n-1}\}$ now consists of a π^* path starting in (w, s_k) visiting states of the form (u, s_k) an arbitrary number of times, but never touching on states with action component s_k or higher in between, and ending in (v, s_k), followed by a π^* path from (v, s_k) to (w', s_j) that does not pass through any pairs (v, s) with $s \in \{s_k, \ldots, s_{n-1}\}$. By the induction hypothesis, a path from (w, s_k) to (v, s_k) of the first kind exists iff $(w, v) \in [\![(K^A_{kkk}(\pi))^*]\!]^{\mathbf{M}}$. Again by the induction hypothesis, a path from (v, s_k) to (w', s_j) of the second kind exists iff $(v, w') \in [\![K^A_{kjk}]\!]^{\mathbf{M}}$. Thus, the required path from (w, s_i) to (w', s_j) in $\mathbf{M} \otimes A$ exists iff $(w, w') \in [\![(K^A_{kkk}(\pi))^*; K^A_{kjk}(\pi)]\!]^{\mathbf{M}}$ iff $(w, w') \in [\![K^A_{ij(k+1)}(\pi)]\!]^{\mathbf{M}}$.

The other two cases are similar. □

The Kleene path lemma is the key ingredient in the following program transformation lemma.

Lemma 2 (Program Transformation) *Assume A has n states $s_0, \ldots, s_{n-1}$. Then:*

$$\mathbf{M} \models_w [A, s_i][\pi]\varphi \text{ iff } \mathbf{M} \models_w \bigwedge_{j=0}^{n-1} [T^A_{ij}(\pi)][A, s_j]\varphi.$$

Proof. Induction on the complexity of π.

Basis, epistemic link case:

$$
\begin{aligned}
& \mathbf{M} \models_w [A, s_i][a]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \text{pre}(s_i) \text{ implies } \mathbf{M} \otimes A \models_{(w,s_i)} [a]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \text{pre}(s_i) \text{ implies for all } s_j \in A, \text{ all } w' \in \mathbf{M}: \\
& \text{if } s_i \xrightarrow{a} s_j, w \xrightarrow{a} w', \text{ then } \mathbf{M} \models_{w'} [A, s_j]\varphi \\
\text{iff} \quad & \text{for all } s_j \in A: \text{ if } s_i \xrightarrow{a} s_j \text{ then } \mathbf{M} \models_w [\text{pre}(s_i); a][A, s_j]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \bigwedge_{j=0}^{n-1} [T_{ij}^A(a)][A, s_j]\varphi.
\end{aligned}
$$

Basis, test case:

$$
\begin{aligned}
& \mathbf{M} \models_w [A, s_i][?\psi]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \text{pre}(s_i) \text{ implies } \mathbf{M} \otimes A \models_{(w,s_i)} [?\psi]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \text{pre}(s_i) \text{ implies } \mathbf{M} \otimes A \models_{(w,s_i)} \psi \to \varphi \\
\text{iff} \quad & \mathbf{M} \models_w \text{pre}(s_i) \text{ and } \mathbf{M} \otimes A \models_{(w,s_i)} \psi \text{ imply } \mathbf{M} \otimes A \models_{(w,s_i)} \varphi \\
\text{iff} \quad & \mathbf{M} \models_w [?(\text{pre}(s_i) \wedge [A, s_i]\psi)][A, s_i]\varphi \\
\text{iff} \quad & \mathbf{M} \models_w \bigwedge_{j=0}^{n-1} [T_{ij}^A(?\psi)][A, s_j]\varphi.
\end{aligned}
$$

Induction step, cases $\pi_1; \pi_2$ and $\pi_1 \cup \pi_2$ are straightforward. The case of π^* is settled with the help of the Kleene path lemma. □

4 Reduction Axioms for Update PDL

The program transformations can be used to translate Update PDL to PDL, as follows:

$$\begin{aligned}
t(\top) &= \top \\
t(p) &= p \\
t(\neg\varphi) &= \neg t(\varphi) \\
t(\varphi_1 \wedge \varphi_2) &= t(\varphi_1) \wedge t(\varphi_2) \\
t([\pi]\varphi) &= [r(\pi)]t(\varphi) \\
t([A,s]\top) &= \top \\
t([A,s]p) &= t(\mathrm{pre}(s)) \rightarrow p \\
t([A,s]\neg\varphi) &= t(\mathrm{pre}(s)) \rightarrow \neg t([A,s]\varphi) \\
t([A,s](\varphi_1 \wedge \varphi_2) &= t([A,s]\varphi_1) \wedge t([A,s]\varphi_2) \\
t([A,s_i][\pi]\varphi &= \bigwedge_{j=0}^{n-1} [T^A_{ij}(r(\pi))]t([A,s_j]\varphi) \\
t([A,s][A',s']\varphi &= t([A,s]t([A',s']\varphi))
\end{aligned}$$

$$\begin{aligned}
r(a) &= a \\
r(B) &= B \\
r(?\varphi) &= ?t(\varphi) \\
r(\pi_1;\pi_2) &= r(\pi_1);r(\pi_2) \\
r(\pi_1 \cup \pi_2) &= r(\pi_1) \cup r(\pi_2) \\
r(\pi^*) &= (r(\pi))^*.
\end{aligned}$$

The correctness of this translation follows from direct semantic inspection, using the program transformation lemma for the translation of $[A,s_i][\pi]\varphi$ formulas. The translation points the way to appropriate reduction axioms, as follows.

Take all axioms and rules of PDL [20, 10, 16], plus the following reduction axioms:

$$\begin{aligned}
[A,s]p &\leftrightarrow (\mathrm{pre}(s) \rightarrow p) \\
[A,s]\neg\varphi &\leftrightarrow (\mathrm{pre}(s) \rightarrow \neg[A,s]\varphi) \\
[A,s](\varphi_1 \wedge \varphi_2) &\leftrightarrow ([A,s]\varphi_1 \wedge [A,s]\varphi_2) \\
[A,s_i][\pi]\varphi &\leftrightarrow \bigwedge_{j=0}^{n-1} [T^A_{ij}(\pi)][A,s_j]\varphi.
\end{aligned}$$

and necessitation for action model modalities. The reduction axioms for $[A,s]p$, $[A,s]\neg\varphi$ and $[A,s](\varphi_1 \wedge \varphi_2)$ are as in [13]. The final reduction axiom is based on program transformation and is new.

If updates with multiple pointed action models are also in the language, we need the following additional reduction axiom:

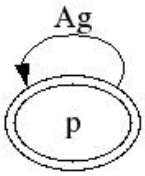
$$[A,S]\varphi \leftrightarrow \bigwedge_{s\in S} [A,s]\varphi$$

Theorem 3 (Completeness) *If* $\models \varphi$ *then* $\vdash \varphi$.

Proof. The proof system for PDL is complete, and every formula in the language of PDL^{DEL} is provably equivalent to a PDL formula. □

5 Special Cases

Public Announcement and Common Knowledge The action model for public announcement that φ consists of a single state s_0 with precondition φ and epistemic relation $\{s_0 \xrightarrow{a} s_0 \mid a \in Ag\}$. Call this model P_φ.



We are interested in how public announcement that φ affects common knowledge among group of agents B, i.e., we want to compute $[P_\varphi, s_0][B^*]\psi$. For this, we need $T_{00}^{P_\varphi}(B^*)$, which is defined as $K_{001}^{P_\varphi}(B)$.

To work out $K_{001}^{P_\varphi}(B)$, we need $K_{000}^{P_\varphi}(B)$, and for $K_{000}^{P_\varphi}(B)$, we need $T_{00}^{P_\varphi}(B)$, which turns out to be $\bigcup_{b\in B}(?\varphi; b)$, or equivalently, $?\varphi; B$. Working upwards from this, we get:

$$K_{000}^{P_\varphi}(B) = ?\top \cup T_{00}^{P_\varphi}(B) = ?\top \cup (?\varphi; B),$$

and therefore:

$$\begin{aligned} K_{001}^{P_\varphi}(B) &= (K_{000}^{P_\varphi}(B))^* \\ &= (?\top \cup (?\varphi; B))^* \\ &= (?\varphi; B)^*. \end{aligned}$$

Thus, the reduction axiom for the public announcement action P_φ with respect to the program for common knowledge among agents B, works out as follows:

$$\begin{aligned} [P_\varphi, s_0][B^*]\psi &\leftrightarrow [T_{00}^{P_\varphi}(B^*)][P_\varphi, s_0]\psi \\ &\leftrightarrow [K_{001}^{P_\varphi}(B)][P_\varphi, s_0]\psi \\ &\leftrightarrow [(?\varphi; B)^*][P_\varphi, s_0]\psi. \end{aligned}$$

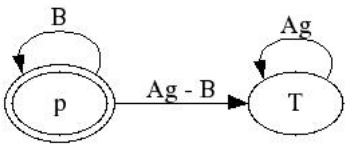
This expresses that every B path consisting of φ worlds ends in a $[P_\varphi, s_0]\psi$ world, i.e., it expresses exactly what is captured by the special purpose operator $C_B(\varphi, \psi)$ introduced in [13].[1]

[1]Indeed, the authors remark in a footnote that their proof system for $C_B(\varphi, \psi)$ essentially follows the usual PDL treatment for the PDL transcription of this formula.

Secret Group Communication and Common Belief The logic of secret group communication is the logic of email CCs. The action model for a secret group message to B that φ consists of two states s_0, s_1, where s_0 has precondition φ and s_1 has precondition $\top$, and where the accessibilities T are given by:

$$T = \{s_0 \stackrel{b}{\to} s_0 \mid b \in B\} \cup \{s_0 \stackrel{a}{\to} s_1 \mid a \in Ag - B\} \cup \{s_1 \stackrel{a}{\to} s_1 \mid a \in Ag\}.$$

The actual world is s_0. The members of B are aware that action φ takes place; the others think that nothing happens. In this thought they are mistaken, which is why CC updates generate KD45 models: i.e., CC updates make knowledge degenerate into belief.



We work out the program transformations that this update engenders for common knowledge among group of agents D. Call the action model CC^B_φ.

We will have to work out $K_{002}^{\mathrm{CC}^B_\varphi} D$, $K_{012}^{\mathrm{CC}^B_\varphi} D$, $K_{112}^{\mathrm{CC}^B_\varphi} D$, $K_{102}^{\mathrm{CC}^B_\varphi} D$.

For these, we need $K_{001}^{\mathrm{CC}^B_\varphi} D$, $K_{011}^{\mathrm{CC}^B_\varphi} D$, $K_{111}^{\mathrm{CC}^B_\varphi} D$, $K_{101}^{\mathrm{CC}^B_\varphi} D$.

For these in turn, we need $K_{000}^{\mathrm{CC}^B_\varphi} D$, $K_{010}^{\mathrm{CC}^B_\varphi} D$, $K_{110}^{\mathrm{CC}^B_\varphi} D$, $K_{100}^{\mathrm{CC}^B_\varphi} D$.

For these, we need:

$$\begin{aligned}
T_{00}^{\mathrm{CC}^B_\varphi} D &= \bigcup_{d \in B \cap D} (?\varphi; d) = ?\varphi; (B \cap D) \\
T_{01}^{\mathrm{CC}^B_\varphi} D &= \bigcup_{d \in D - B} (?\varphi; d) = ?\varphi; (D - B) \\
T_{11}^{\mathrm{CC}^B_\varphi} D &= D \\
T_{10}^{\mathrm{CC}^B_\varphi} D &= ?\bot
\end{aligned}$$

It follows that:

$$\begin{aligned}
K_{000}^{\mathrm{CC}^B_\varphi} D &= ?\top \cup (?\varphi; (B \cap D)) \\
K_{010}^{\mathrm{CC}^B_\varphi} D &= ?\varphi; (D - B) \\
K_{110}^{\mathrm{CC}^B_\varphi} D &= ?\top \cup D, \\
K_{100}^{\mathrm{CC}^B_\varphi} D &= ?\bot
\end{aligned}$$

From this we can work out the K_{ij1}, as follows:

$$\begin{aligned}
K_{001}^{\mathrm{CC}_\varphi^B} D &= (?\varphi; (B \cap D))^* \\
K_{011}^{\mathrm{CC}_\varphi^B} D &= (?\varphi; (B \cap D))^*; (D - B) \\
K_{111}^{\mathrm{CC}_\varphi^B} D &= ?\top \cup D \\
K_{101}^{\mathrm{CC}_\varphi^B} D &= ?\bot.
\end{aligned}$$

Finally, we get K_{002} and K_{012} from this:

$$\begin{aligned}
K_{002}^{\mathrm{CC}_\varphi^B} D &= K_{001}^{\mathrm{CC}_\varphi^B} D \cup K_{011}^{\mathrm{CC}_\varphi^B} D; (K_{111}^{\mathrm{CC}_\varphi^B} D)^*; K_{101}^{\mathrm{CC}_\varphi^B} D \\
&= K_{001}^{\mathrm{CC}_\varphi^B} D \quad \text{(since the righthand expression evaluates to } ?\bot) \\
&= (?\varphi; (B \cap D))^* \\
K_{012}^{\mathrm{CC}_\varphi^B} D &= K_{011}^{\mathrm{CC}_\varphi^B} D \cup K_{011}^{\mathrm{CC}_\varphi^B} D; (K_{111}^{\mathrm{CC}_\varphi^B} D)^* \\
&= K_{011}^{\mathrm{CC}_\varphi^B} D; (K_{111}^{\mathrm{CC}_\varphi^B} D)^* \\
&= (?\varphi; (B \cap D))^*; (D - B); D^*.
\end{aligned}$$

Thus, the program transformation for common belief among D works out as follows:

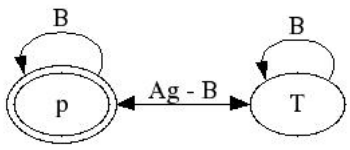
$$[\mathrm{CC}_\varphi^B, s_0][D^*]\psi \quad \leftrightarrow \quad [(?\varphi; (B \cap D))^*][\mathrm{CC}_\varphi^B, s_0]\psi \wedge [(?\varphi; (B \cap D))^*; (D - B); D^*][\mathrm{CC}_\varphi^B, s_1]\psi.$$

Compare [19] for a direct axiomatisation of the logic of CCs.

Group Messages and Common Knowledge The action model for a group message to B that φ consists of two states s_0, s_1, where s_0 has precondition φ and s_1 has precondition $\top$, and where the accessibilities T are given by:

$$T = \{s_0 \xrightarrow{b} s_0 \mid b \in B\} \cup \{s_1 \xrightarrow{b} s_1 \mid b \in B\} \cup \{s_0 \xrightarrow{a} s_1 \mid a \in Ag - B\} \cup \{s_1 \xrightarrow{a} s_0 \mid a \in Ag - B\}.$$

This captures the fact that the members of B can distinguish the φ update from the $\top$ update, while the other agents (the members of $Ag - B$) cannot. The actual action is s_0. Call this model G_φ^B.



A difference with the CC case is that group messages are S5 models. Since updates of S5 models with S5 models are S5, group messages engender common knowledge (as opposed to

mere common belief). Let us work out the program transformation that this update engenders for common knowledge among group of agents D.

We will have to work out $K_{002}^{G_\varphi^B}D$, $K_{012}^{G_\varphi^B}D$, $K_{112}^{G_\varphi^B}D$, $K_{102}^{G_\varphi^B}D$.

For these, we need $K_{001}^{G_\varphi^B}D$, $K_{011}^{G_\varphi^B}D$, $K_{111}^{G_\varphi^B}D$, $K_{101}^{G_\varphi^B}D$.

For these in turn, we need $K_{000}^{G_\varphi^B}D$, $K_{010}^{G_\varphi^B}D$, $K_{110}^{G_\varphi^B}D$, $K_{100}^{G_\varphi^B}D$.

For these, we need:

$$\begin{aligned}
T_{00}^{G_\varphi^B}D &= \bigcup_{d\in D}(?\varphi;d) =?\varphi;D,\\
T_{01}^{G_\varphi^B}D &= \bigcup_{d\in D-B}(?\varphi;d) =?\varphi;(D-B),\\
T_{11}^{G_\varphi^B}D &= D,\\
T_{10}^{G_\varphi^B}D &= D-B.
\end{aligned}$$

It follows that:

$$\begin{aligned}
K_{000}^{G_\varphi^B}D &= ?\top\cup(?\varphi;D),\\
K_{010}^{G_\varphi^B}D &= ?\varphi;(D-B),\\
K_{110}^{G_\varphi^B}D &= ?\top\cup D,\\
K_{100}^{G_\varphi^B}D &= D-B.
\end{aligned}$$

From this we can work out the K_{ij1}, as follows:

$$\begin{aligned}
K_{001}^{G_\varphi^B}D &= (?\varphi;D)^*,\\
K_{011}^{G_\varphi^B}D &= (?\varphi;D)^*;?\varphi;D-B,\\
K_{111}^{G_\varphi^B}D &= ?\top\cup D\cup(D-B;(?\varphi;D)^*;?\varphi;D-B),\\
K_{101}^{G_\varphi^B}D &= D-B;(?\varphi;D)^*.
\end{aligned}$$

Finally, we get K_{002} and K_{012} from this:

$$\begin{aligned}
K_{002}^{G_\varphi^B}D &= K_{001}^{G_\varphi^B}D\cup K_{011}^{G_\varphi^B}D;(K_{111}^{G_\varphi^B}D)^*;K_{101}^{G_\varphi^B}D\\
&= (?\varphi;D)^*\cup\\
&\quad (?\varphi;D)^*;?\varphi;D-B;(D\cup(D-B;(?\varphi;D)^*;?\varphi;D-B))^*;D-B;(?\varphi;D)^*,\\
K_{012}^{G_\varphi^B}D &= K_{011}^{G_\varphi^B}D;(K_{111}^{G_\varphi^B}D)^*\\
&= (?\varphi;D)^*;?\varphi;D-B;(D\cup(D-B;(?\varphi;D)^*;?\varphi;D-B))^*.
\end{aligned}$$

Abbreviating $D \cup (D - B; (?\varphi; D)^*; ?\varphi; D - B)$ as π, we get the following transformation for common knowledge among D after a group message to B that φ:

$$\begin{array}{rcl} [G^B_\varphi, s_0][D^*]\psi & \leftrightarrow & [(?\varphi; D)^* \cup ((?\varphi; D)^*; ?\varphi; D - B; \pi^*; D - B; (?\varphi; D)^*)][G^B_\varphi, s_0]\psi \\ & & \wedge \\ & & [(?\varphi; D)^*; ?\varphi; D - B; \pi^*][G^B_\varphi, s_1]\psi. \end{array}$$

6 Conclusion

That any dynamic epistemic update with a finite action model can be expressed in PDL is not new. [4] has a theorem to that effect, and it also follows from the reduction of dynamic epistemic updating to automata PDL from [13]. What is new is the constructive definition of the embedding of dynamic epistemic updates in PDL.

Does it follow from the reduction to PDL that dynamic epistemic logic is essentially 'nothing but' PDL? If one takes dynamic epistemic logic to be the result of adding generic epistemic updating with finite action models to PDL, then the answer is 'yes'. If one takes dynamic epistemic logic to be the result of adding updates with finite action models to a base multimodal logic enriched with common knowledge operators, then the answer is: maybe not. It depends on whether PDL is more expressive than *this*. This technical question is still open, as far as I know.

The program transformation approach to epistemic updates makes clear that it is possible to view the update actions themselves as a kind of finite automata. The similarity is most striking in the definition of the transformation for starred programs. The definition of transformed $\pi^\star$ paths in terms of operators $K_{ijk}(\pi)$ closely resembles the definition of sets of regular languages L_k generated by moving through a nondeterministic finite automaton without passing through states numbered k or higher, in the well-known proof of the fact that languages generated by nondeterministic finite automata are regular [15, Thm 2.5.1].

Finally, note that our program transformations transform regular programs to regular programs, i.e., they act as transducers. The effect of transducers can be specified with regular relations, so a version of PDL with regular relations might be an appropriate way of getting rid of update models in the epistemic language altogether. E.g., the effect of the public update with φ is captered by the regular relation $\bigcup_{a \in Ag}(a ::?\varphi; a)$, where :: denotes replacement.

Acknowledgement Thanks to Alexandru Baltag, Johan van Benthem, Barteld Kooi, Larry Moss and Ji Ruan for helpful comments and inspiring discussion.

References

[1] Baltag, A. A logic for suspicious players: epistemic action and belief-updates in games. *Bulletin of Economic Research 54*, 1 (2002), 1–45.

[2] Baltag, A., and Moss, L. Logics for epistemic programs. *Synthese 139*, 2 (2004), 165–224.

[3] Baltag, A., Moss, L., and Solecki, S. The logic of public announcements, common knowledge, and private suspicions. Tech. Rep. SEN-R9922, CWI, Amsterdam, 1999.

[4] Baltag, A., Moss, L., and Solecki, S. The logic of public announcements, common knowledge, and private suspicions. Tech. rep., Dept of Cognitive Science, Indiana University and Dept of Computing, Oxford University, 2003.

[5] Benthem, J. v. Language, logic, and communication. In *Logic in Action*, J. van Benthem, P. Dekker, J. van Eijck, M. de Rijke, and Y. Venema, Eds. ILLC, 2001, pp. 7–25.

[6] Benthem, J. v. One is a lonely number: on the logic of communication. Tech. Rep. PP-2002-27, ILLC, Amsterdam, 2002.

[7] Ditmarsch, H. v. *Knowledge Games*. PhD thesis, ILLC, Amsterdam, 2000.

[8] Eijck, J. v. Dynamic epistemic modelling. manuscript, CWI, Amsterdam, 2004.

[9] Fagin, R., Halpern, J., Moses, Y., and Vardi, M. *Reasoning about Knowledge*. MIT Press, 1995.

[10] Fischer, M. J., and Ladner, R. E. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences 18*, 2 (1979), 194–211.

[11] Gerbrandy, J. *Bisimulations on planet Kripke*. PhD thesis, ILLC, 1999.

[12] Harel, D., Kozen, D., and Tiuryn, J. *Dynamic Logic. Foundations of Computing*. MIT Press, Cambridge, Massachusetts, 2000.

[13] Kooi, B., and van Benthem, J. Reduction axioms for epistemic actions. Manuscript, Groningen/Amsterdam, 2004.

[14] Kooi, B. P. *Knowledge, Chance, and Change*. PhD thesis, Groningen University, 2003.

[15] Lewis, H., and Papadimitriou, C. *Elements of the Theory of Computation*. Prentice-Hall, 1981.

[16] Parikh, R. The completeness of propositional dynamic logic. In *Mathematical Foundations of Computer Science 1978*. Springer, 1978, pp. 403–415.

[17] Pratt, V. Semantical considerations on Floyd–Hoare logic. *Proceedings 17th IEEE Symposium on Foundations of Computer Science* (1976), 109–121.

[18] Pratt, V. Application of modal logic to programming. *Studia Logica 39* (1980), 257–274.

[19] Ruan, J. Exploring the update universe. Master's thesis, ILLC, Amsterdam, 2004.

[20] Segerberg, K. A completeness theorem in the modal logic of programs. In *Universal Algebra and Applications*, T. Traczyck, Ed. Polish Science Publications, 1982, pp. 36–46.